A MULTI-AGENT SYSTEM FOR INTELLIGENT MONITORING OF AIRLINE OPERATIONS

António Castro [a,b] Eugénio Oliveira [b]

[a] *TAP Portugal, Lisbon Airport*
Ed. 27 Sul Piso 6 Sala 27, Lisboa, Portugal
[b] *LIACC-NIAD&R, Faculty of Engineering, University of Porto*
R. Dr. Roberto Frias, 4200-465 Porto, Portugal

Abstract

There are quite a few solutions for crew scheduling, including some commercial applications. The same happens for aircraft scheduling and for flight scheduling including revenue management. However, the airline operations problem did not receive the same attention as the other airline scheduling problems. In this paper we introduce this problem and report the work we are doing in the development of a Distributed Multi-Agent System that will encompass tasks like crew and aircraft recovery among others that are typical of airline operations control. The MAS deals with different operational bases and all bases cooperate to find the solutions to the local problems. Robustness is a key feature and we achieve that through redundancy in finding the possible solutions to the problem, using agents that compete in finding for the best solution to be applied. To be an "Intelligent System" some kind of learning must be available. We are using learning to define the crew member's profile, to learn the use of stand by crew members and include this learning in future crew scheduling and in suggesting new solutions based on previous decisions. Finally, we would like to explore the possibility of having a "kind of electronic market" for available crew members/aircrafts among airline companies, to be used in crew and aircraft recovery. This would work as a "market" of solutions to specific local problems and these solutions would compete with the recommended local solutions. To develop the system the latest MAS methodologies, frameworks, tools and technologies will be used. This includes GAIA, JADE, Agent-web services and IBM Rational suite of tools.

1 Introduction

One of the most important areas in an airline company is the crew scheduling. Basically, after defining the commercial flight schedule it is necessary to assign the necessary crew members to the flights (and before that the aircraft fleet) and produce what's called a roster. Usually this is done in two phases: (1) pairing construction, that is, the creation of anonymous pairings starting and ending at home base that should cover all the flight positions defined in the company flight schedule, and (2) assign all pairings and other activities (for example, stand by duties) to individual crew members. This is known as crew rostering. Traditionally, airline crew scheduling problems are solved using Operations Research (OR) techniques. The paper [1] gives an overview of OR applications in the air transport industry.

After publishing the crew schedule or roster it is necessary to monitor any changes to this schedule and act accordingly. The same happens with the aircraft schedule. These tasks are performed in the operation phase. Our MAS addresses this important area, especially with the inclusion of intelligent features that allows finding the best solution to the majority of the problems that arise during airline operations. This includes the replacement of crew members that do not report for duty, flight delays, commercial changes, and so on. This kind of action is known by crew and aircraft recovery. According to [11], most of the work of operation recovery (or disruption management) focuses on recovering a particular resource, either aircrafts, crew or passengers. In [6], the authors state that "the requirements for a tool as seen from the airline companies are, however, still substantially different from the services offered by commercial tools, and from the performance seen in all the prototype tools proposed in the literature." In our proposal we try to do a step forward in building a system that will encompass all tasks usually performed in the airline operations phase. We believe that a MAS is an adequate approach to deal with this subject. In our MAS the crew and aircraft recovery process, that usually are decision support tools in the systems found in the literature, are agents that, with a great degree of autonomy, will try to achieve their objectives.

The MAS is also able to deal with different operational bases, with their own resources, solving the local problems and contributing to the solution of problems in other bases. We would also like to address the topic of cooperation between airlines in recovery operation. According to [11], "research on the recovery operation to this date only deals with a single airline. Cooperation between airlines is not supported." We are working in a definition of a Electronic Market of crew members and aircrafts, that will contribute to the cooperation between airlines companies in this area.

During the operations phase there are a lot of things that the system can learn: (1) the individual crew member profile related with bids preferences and individual execution of the schedule roster, (2) the real utilization of stand by crew members and (3) new solutions based on the previous decisions applied to solve the problems. The result of this learning can be applied during crew and aircraft recover (3) and during the crew scheduling (1) (2) to improve the resulting crew schedule.

To do our MAS we are observing the Airline Operations Department of TAP Portugal [14] - the major Portuguese airline. This department works 24 hours a day and 7 days a week. We have access to all the information and we are allowed to talk with the users and find out how they solve the problems.

The rest of this paper is structured in the following way. Section 2 discusses the Airline Scheduling in a more detailed way. Section 3 introduces the methodologies, tools and technologies used. Section 4 presents the Vision and Scope of our proposed MAS. Section 5 presents the analysis and architecture of the MAS and Section 6 gives an overview of what we want to explore in the future.

2 The Airline Scheduling Problem

According to [11] the interdependences of the several phases in the airline scheduling problem are illustrated according to figure 1.

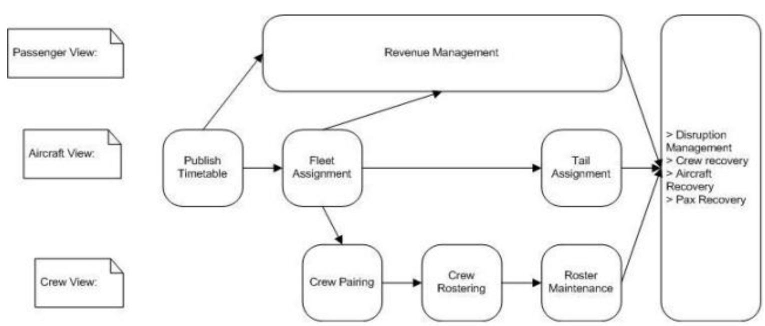

Figure 1: Phases and interdependences of the scheduling problem

Publish Timetable: The commercial flight schedule, that is, the schedule of flights that can be "sold" to passengers.

Fleet Assignment: The allocation of aircraft type (fleets) to the schedule flights.

After these two phases the planning process runs in parallel. The crew scheduling problem is usually solved in two phases: crew pairing and crew rostering.

Crew Pairing : Consists of creating anonymous pairings, starting and ending at home base. Legal rules are applied at this time (for example, maximum duration of each flight duty period, minimum rest time between two flight duty periods, maximum consecutive critical periods, etc.) as well as some company rules (for example, minimum time for the rotation of the aircraft, maximum number of flight legs in each flight duty period, maximum number of duration days for each pairing, etc.).

Crew Rostering: Consists of assigning the pairings and other activities (for example, stand by duties, training, and reserves) and applying legal rules (for example, days off, vacation, and limits on duty/block time) to individual crew members. For a more detailed list please consult [10]. The crew scheduling must be completed a few weeks before the day corresponding to the start of the schedule and the result of this phase is a personalized roster for each crew member, usually for a one month period.

Roster Maintenance: Consists of reflecting later changes, i.e., commercial flights changes, roster availability, etc., in the crew schedule.

Tail Assignment: Consists of assigning the actual aircrafts (tail numbers) to flights and, consequently, the routing of aircrafts individuals. This is typically done a few days before day of operations. In TAP Portugal [14], for example, the personal roster is already published with the aircraft tail numbers. However, this assignment is not final and it is dependent of the operations control.

Revenue Management: Corresponds to the adjustment of prices and seat availability according to the market and is carried out during the entire period from the publication of the flight timetable to the day of operations.

Operations Control: Or Disruption Management as called in [11] corresponds to the monitoring of the schedule execution, trying to solve all the problems related with crew members, aircrafts and passengers. Every change in the schedule, for example, flight delay or cancellation, aircraft fleet change, crew members that do not report for duty, new flights scheduled, etc., that happens in this phase must be feasible for crew as well as for aircraft and should minimize the passenger inconvenience. A high-level view of this process as found in [11] is presented in figure 2.

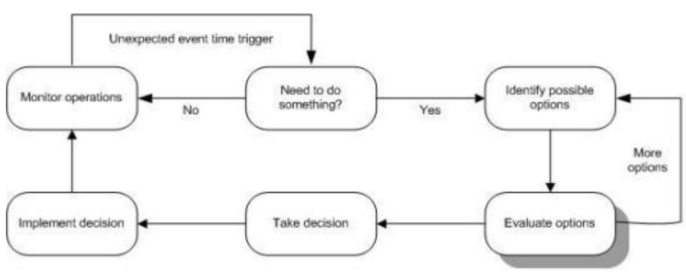

Figure 2: Disruption management

During this phase, we call Crew Recovery to the process of solving crew related problems, Aircraft Recovery to the process of solving aircraft related problems and Passenger Recovery to the process of solving passenger problems. We will focus our interest, especially, in the Crew Recovery process and, until a certain extension, in the Aircraft Recovery process. We will not address the Passenger Recovery process in this paper. Traditionally, the aircraft crew scheduling problem and the operations recovery problem (or disruption management) are approached using OR methods and tools. For an overview of applications of OR in the air transport industry we recommend the reading of [1]. For a review of the concepts, models and methods used in disruption management in the airline industry, we recommend the reading of [6].

3 Methodology, Tools and Technologies

To develop our MAS we adopted the use of GAIA [16] as the Analysis and Design methodology. Figure 3 gives a commented overview of this methodology.

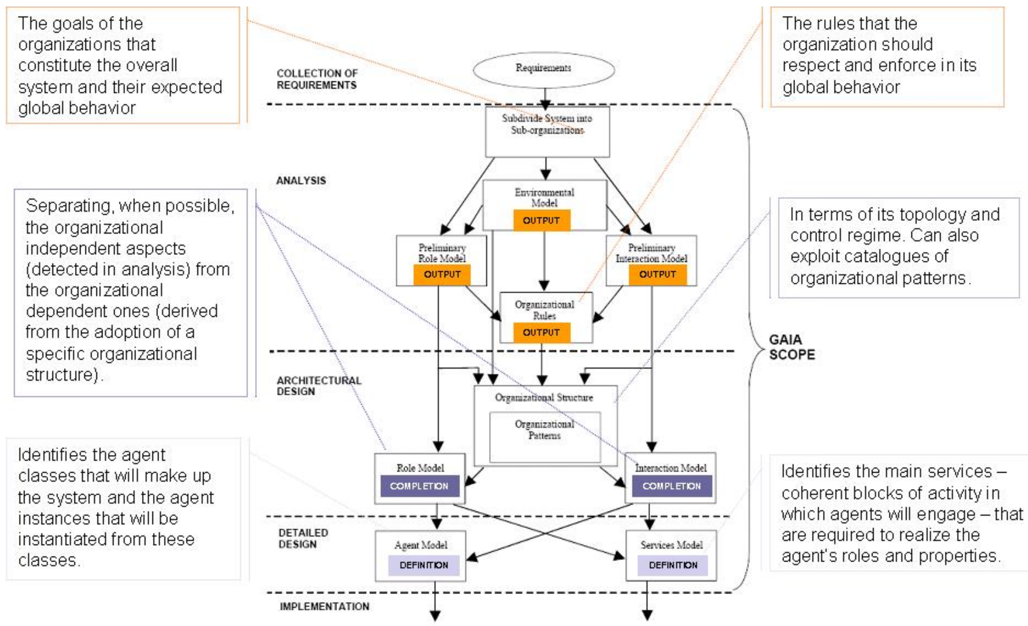

Figure 3: GAIA Methodology: a commented overview

In the analysis phase this methodology produces the following outputs: The environment model, the preliminary role model, the preliminary interaction model and the organizational rules. In the architectural design the role and interaction model are finished and in the detailed design the Agent Model and the Services Model are defined. GAIA does not deal directly with implementation issues. The results of GAIA are a detailed but technology-neutral specification.

Following the suggestions of the authors of GAIA and because the collection of requirements are an input for GAIA, we have adopted a goal-driven early-requirements analysis [13], [5], [4] for early-requirements analysis and AUML [2] as the notation to use when appropriate. One may ask why we did not adopt a complete methodology like TROPOS [4], instead of using GAIA that does not support all phases. From a strict business perspective it would be better to use a more complete methodology. However, from an academic perspective, we are also interested in see how we can complement GAIA with the missing phases. We hope to collect some interesting data on this subject.

To model and develop the MAS we use IBM Rational Software Architect [15] and, for the database, IBM Universal DB2. IBM Rational Software Architect is being used as a modeling tool and as a development tool. In modeling we are using the TAOM4E - Tool for Agent Oriented visual Modeling for the Eclipse Platform [7] plug-in, to do the early requirements phase according to TROPOS. This tool supports a model-driven, agent oriented software development and, in particular, the TROPOS methodology. We are also using IBM Rational Software Architect for analysis and design due to the fact that it supports UML 2.0. We are following the approach from Bernhard Bauer and James Odell paper "UML 2.0 and Agents: How to build agent-based systems with the new UML standard" [3].

Finally we use Java Agent Development Framework [8] as the middleware platform for our system. JADE is a pure Java, middleware platform intended for the development of distributed multi-agent applications based on peer-to-peer communication. Includes Java classes to support the development of applications agents, and the "run-time" environment that provides the basic services for agents to execute. We will use IBM Rational Software Architect to develop our system using JAVA [12] and JADE.

4 Vision and Scope of our MAS

As we stated in the introduction of our paper, the problems related to airline scheduling and operations are, traditionally, solved using Operations Research techniques. We believe that, using MAS technology, we can go a step forward and create a more autonomous system without sacrifying the demanding commercial and operational requirements of this domain. What distinguishes our approach from Operations Research traditional approaches is the wider scope of our MAS. We want the users to be managers and not controllers. In our MAS the decision support tools of the OR techniques will be agents that will compete and/or collaborate to find the best solution to the problems that may appear during airline operations control. Finally, our MAS will take advantage of the geographic distribution of resources through operational bases. As a summary, we are developing a Distributed Multi-agent System, with emphasis in the following:

- Monitoring events.
- Automate the resolution of the most trivial problems in Crew and Aircraft Recovery.
- Solve problems for each operational base and share available resources with other bases.
- Open to solutions from other airline companies.
- Learning preferences and new solutions.
- Robustness through redundancy [9].

The main problems that arise during operations control are the following (for a more detailed list of typical problems consult [10]):

Flight delays: Due to weather conditions, air traffic control restrictions, boarding problems, etc. This type of problems can lead to delays on the flights that depend on the arrival of the aircraft and/or in connection flights. Crew problems might arise.

Aircrafts malfunctions: Can lead to problems similar to the previous one. Crew and aircraft problems might arise.

Crew members not reporting for duty: It is necessary to find an available crew member for replacement. Can lead to flight delays in some conditions or, in the most complicated cases, to flight cancellation.

Crew member's delays in reporting for duty: Might be necessary to call the stand by crew. Might lead to problems similar to the previous one.

Commercial changes to the flights: Changes in the flight schedule, new flights, cancellation of flights and so on. Usually leads to a crew scheduling (pairing and rostering) problem.

The final goals of our MAS are:

Problem anticipation: A successful and effective monitoring of events should be able to forecast possible problems and allow on time resolutions.

Detailed and correct information: The system should be able to give the necessary information for the users to understand how the solutions were achieved or, in some rare cases, allow the users to find a solution.

Apply the solution: The system should be able to apply automatically the best solution found or, in some especial cases, the solution selected by the user. This might include the automatic creation of crew pairings and their assignment to crew members.

Crew member profile: The system should be able to learn the crew members bid preferences as well the individual execution of the crew roster.

Use of the stand by crew members: The system should be able to learn the utilization of the stand by crew members during operations control. The knowledge that came up from this learning should be used in future crew scheduling. This will allow a better use of this rare and expensive resource.

Quick solution for trivial problems: Crew problems like the replacement of a crew member and/or flight delays that lead to crew problems, should be solved quickly and automatically.

From controllers to managers: The system should allow a reduction of the number of users that usually take care of operations control. The idea is to have Operations Managers/Crew Tracker Managers instead of Operations Controller and Crew Tracker Controllers. The system should do the control and the users should be the managers.

Robustness through redundancy [9]: The system should be able to find solutions to the same problem using different algorithms.

As one can see this domain has very complex problems. Our objectives are ambitious but also realistic. To be able to achieve them we came up with the following list of features to be implemented:

- Monitoring crew members reporting for duty.
- Monitoring of flight departures and arrivals.
- Monitoring stand by crew members.
- Monitoring commercial changes to the flight schedule.
- Log absences communicated by crew members to the operations control.
- Implement several algorithms to be used in finding the solutions: Evolutionary Computing, Column Generation, Integer Programming, Heuristic Methods and others.
- Assign automatically the best crew to open positions in a pairing, after choosing the best solution from the several that have been proposed by the different algorithms.
- Create/change pairings resulting from commercial flight changes or other problems, after choosing the best solution from the several that have been proposed by the different algorithms.
- Log and learn the profile of each crew member related with bid preferences to be used in future crew scheduling.
- Learn the profile of each crew member from the individual execution of each crew member's schedule and apply that profile in future crew scheduling.
- Learn from the use of the stand by crew members and apply that knowledge in future crew scheduling.
- Learn from the applied solutions and with that knowledge propose new solutions.

5 Analysis and Proposed Architecture

After observing the actual airline control operations system at TAP and combing that observation with the vision and scope for the system, we came up with some hypothesis and predictions. The experiments regarding our hypothesis are still "work in progress". However we think it is important to write our hypothesis and predictions now:

First, we hypothesize that the main objective of airline operation (flights always on time) will be much easier to achieve (less flights delayed) if we take advantage of the fact that the crew members belong to different bases. We predict that if we solve the problems first with local resources and then with non-local resources, the solutions to the eventual problems will be much faster to find and, in some cases, the non-local solutions may be the only solution available.

Second, we hypothesize that the use of different algorithms to solve the same problem (in crew and aircraft recovery) will improve the achievement of the main objective of the crew and aircraft recovery process (to always find the better solution regarding "ensuring every flight has a crew" and "ensuring that all flights are on time", respectively). We predict that using different algorithms (genetic algorithms, heuristic, etc.) in comparison with using always the same algorithm, to solve the same problem, will permit (1) to always find the best solution (according to the criteria defined by the company) and (2) to always find a solution, especially taking into account the fact that we might benefit from solutions presented by other bases, as stated in the first hypothesis.

Third, we hypothesize that the implementation of a learning mechanism that will learn from the use of crew members (in comparison with the previous and published schedule and in characterizing specific situations) will permit a better use of the resources (especially crew members) in future schedules. We predict, for example, that if we learn the real use of stand by crew members in each month and in specific situations, it will allow adjusting the stand by roster in similar months or similar situations of future schedules, permitting to release crew members to be schedule to flights (crew members are one of the most expensive resources in an airline company).

Fourth, we hypothesize that if we extend the learning mechanism to learn the profile of each crew member, regarding his/her preferences, it will increase the level of satisfaction of them. We predict that applying the learned profile of each crew member in future schedules of corresponding months it will produce a roster that will achieve the goals of the airline company and, at the same time, the satisfaction of the crew members. Increasing the level of satisfaction of a crew member will decrease the crew member's absence to work.

In this paper our main focus is the two first hypotheses. The examples that we provide will be more concentrated in the crew recovery process. The other two processes (aircraft and passenger recovery) will be presented only when necessary to understand the full picture.

As stated before, for the early-requirements we use a goal-driven approach, in this case, based on TROPOS [4]. The first step is to do the Actor Modeling, which, according to [4], "consists of identifying and analyzing both the actors of the environment and the system's actors and agents". In early requirements modeling the main focus is on the application domain stakeholder's intentions as "social actors which want to achieve goals". The actor diagram is a representation of that modeling, and a simplified version for airline operations is presented in figure 4.

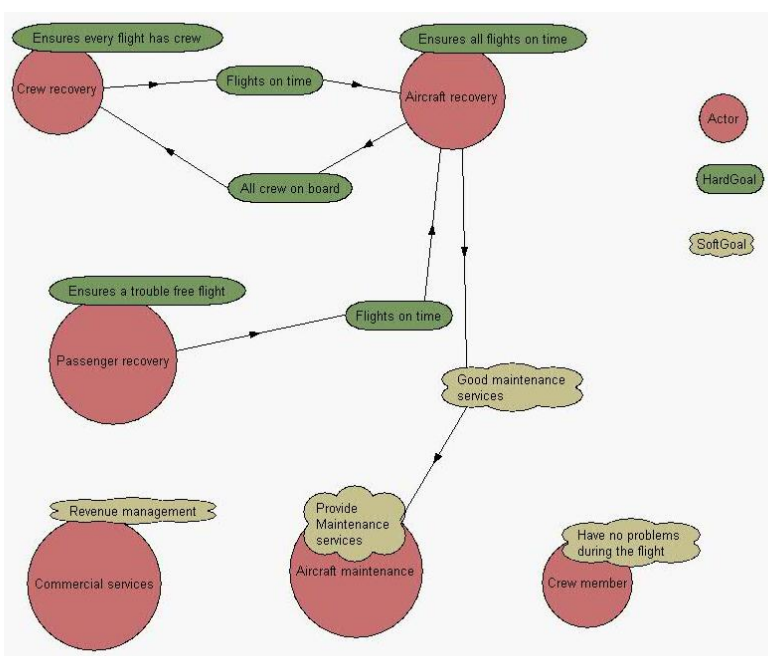

Figure 4: Simplified actor diagram for airline operations control in TAP

The Crew Recovery actor has the main objective of ensuring that every flight has all the necessary crew members to operate. This is indicated in this diagram by the hardGoal represented above the actor. The Aircraft Recovery actor has the main objective of ensuring that all flights departure on time. The objectives of these two actors are related by two hard goal dependencies: crew recovery actor depends on aircraft recovery actor to fulfill it's objectives by the *flights on time* goal dependency. Likewise, aircraft recovery actor depends on crew recovery actor to fulfill it's objectives by the *all crew on board* goal dependency and on aircraft maintenance actor to have *good maintenance services*. The Passenger Recovery actor has the main objective of ensuring a trouble free flight to all passengers and this objective is pending by the *flights on time* goal dependency with the aircraft recovery actor. All these three actors have very precise objectives (that's the reason for being indicated as a hardgoal) and their objectives are much related to each other.

From this diagram we can also observe other actors: Commercial Services actor is preoccupied with doing the best revenue management possible. Aircraft Maintenance Services want to provide the best maintenance services and contributing to the fulfillment of the aircraft recovery actor objectives through the *good maintenance services* soft goal dependency. Finally, the Crew Member actor just wants no problems during the duty.

The next step is to do Goal Modeling, "conducted from the point of view of the actor, by using three basic reasoning techniques: means-end analysis, contribution analysis, and, AND/OR decomposition. In particular, means-end analysis aims at identifying plans, resources and softgoals that provide means for achieving a goal. Contribution analysis identifies goals that can contribute positively or negatively in the fulfillment of the goal to be analyzed. In a sense, it can be considered as an extension of means-end analysis, with goals as means. AND/OR decomposition, combines AND and OR decompositions of a root goal into sub-goals, modeling a finer goal structure." [4]. The goal diagram is a representation of this modeling and in figure 5 we present a simplified version of the Crew Recovery Goal Diagram.

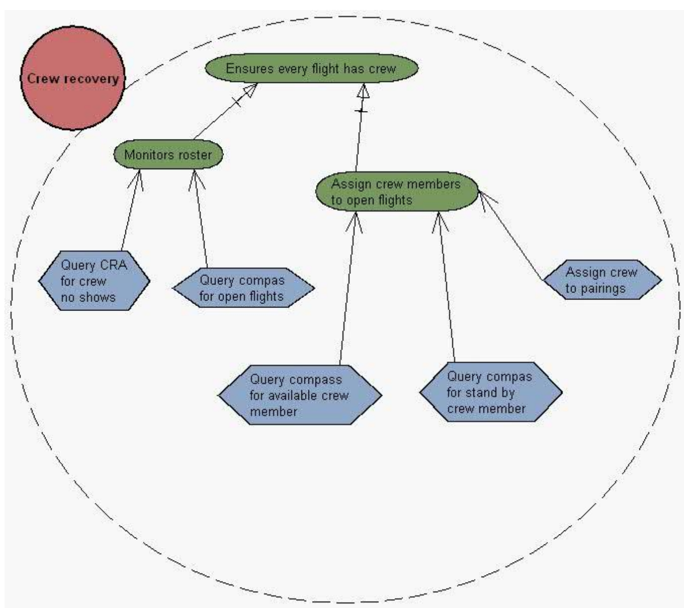

Figure 5: Simplified version of the Crew Recovery Goal Diagram

For the actor Crew Recovery the goal *ensures every flight as a crew* is decomposed into *monitor's roster* and *assigns crew members to open flights*. These two goals must be achieved so that *ensures every flight has crew* will also be fulfilled (that's the reason for the AND-decomposition). Goal decomposition can be closed through a means-end analysis with the objective of identifying plans, resources and softgoals that provides means to achieve the goal. For example, the plan *Query CRA for crew no shows* (depicted as a hexagon) and *Query compass for open flights* are means to fulfill the goal *monitors roster*.

In figure 6 we present a simplified version of the Aircraft Recovery Goal Diagram.

These are only simplified versions of some of the diagrams of the early-requirements of our

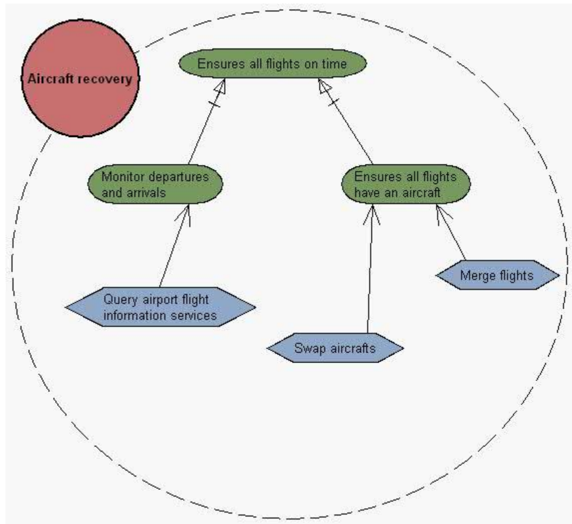

Figure 6: Simplified version of the Aircraft Recovery Goal Diagram

MAS. The final result of this phase is a set of dependencies among actors. These dependencies were built doing goal and plan analysis on each goal until all goals have been analyzed. To start the analysis phase and according to GAIA [16], we need to have a collection of requirements. The results from the early requirements phase, that presents the actors' involved, their roles and goals, are very helpful for the analysis phase. As a summary and for the analysis phase, we need to do the following tasks:

- Determine whether multiple organizations have to coexist in the system and became autonomous interacting MAS. The use of a goal oriented requirements analysis helps in identifying the conditions that will help to decide if it is necessary or not to have multiple organizations.
- Build the Environment Model, treating it in terms of "abstract computational resources such as variables or tuples, made available to the agents for sensing, effecting or consuming".
- Build the Preliminary Role Model, identifying the basic skills required by the organization to achieve its goals, as well as the basic interactions that are required for the exploitation of these skills. Due to the use of a goal oriented early requirements analysis, this identification activities are facilitated because we have modeled the characteristics of the system in terms of actors involved and their goals.
- Build the Preliminary Interaction Model, capturing the dependencies and relationships between the various roles in the MAS organization, in terms of one protocol definition for each type of inter role interaction. From the goal oriented early requirements phase, it is possible to have an idea of the necessary role interactions and their dependencies and relationships.
- Define the Organizational Rules, that is, the responsibilities of the organization as a whole.

The diagram of figure 7 presents a simplified version of our MAS architecture, after doing the analysis according to GAIA. To avoid filling the diagram we did not represent all the resources included in the environment as well as other agents that might be needed. We have also opted to use the same graphical that appears in the GAIA paper, instead of a more formal one as AUML.

From this diagram we can see that there are two sub-organizations inside our MAS. According to GAIA, sub-organizations are necessary when there are portions of the overall system that have one or more of these conditions:

- Exhibit a behavior specifically oriented towards the achievement of a given sub-goal.
- Interact loosely with other portions of the system.

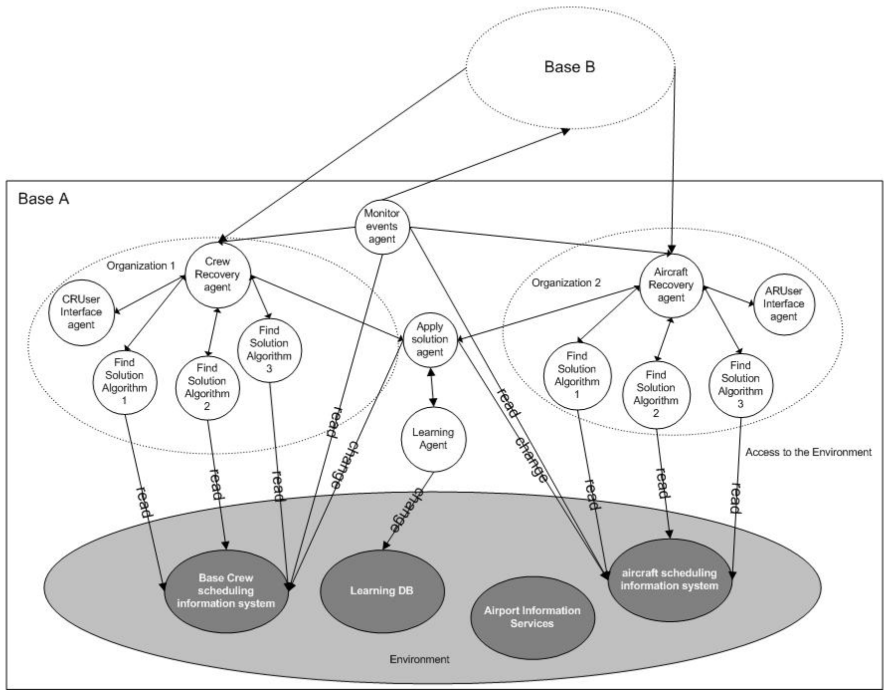

Figure 7: A simplified version of the proposed architecture for the MAS organization

- Require competences that are not needed in other parts of the system.

As one can see, the two sub-organizations defined in this diagram, sub-organization 1 and sub-organization 2, have a behavior specially oriented to the achievement of a given sub-goal, that is, fulfill the objective “ensures every flight has crew” and “ensures all flights on time”, respectively. It is also possible to infer from this diagram that each one of these sub-organizations are hierarchical, having the Crew Recovery Agent and Aircraft Recovery Agent on the top of that hierarchy. Here the agents Find Solution 1, Find Solution 2 and Find Solution 3, are competing so that they present the best solution to the Crew Recovery agent. A similar situation happens on the other sub-organization. With this organization we hope to prove our second prediction and give “robustness” to our MAS through this “redundancy”.

It is necessary to point out that other bases can also propose solutions to the problems raised by another base. In this case we have cooperation between the bases because it will be difficult for another base to give a better solution than the ones found by the local base, making it impossible to compete. However, this cooperation will be much important because the solution presented by another base might be the only one possible. With this cooperation we hope to prove our first prediction.

Our third and fourth prediction are related with learning from the utilization of the crew members in comparison with the published roster and in characterizing specific situations, and from learning the profiles and preferences of each crew member, respectively. We hope to improve future schedules from the learning of our systems. In this diagram this is simply identified by the Learning Agent. In a future paper we will develop much more this feature. The Monitor Events Agent will be responsible for monitoring all events that happen on our system, like, for example, crew members not reporting for duty, flight delays, and so on. This agent will be responsible to broadcast for the entire system when such an event happens. The Apply Solution Agent will be responsible to apply the solution when that solution has been found. With this approach and when implementing our system in an airline company, it will be easier to request approval from a user before applying the solution.

From this diagram we can also see the interactions between agents, represented by the arrows connecting them. As an example, it is possible to see that Monitor Events Agent interacts with Crew Recovery Agent and Aircraft Recovery Agent, but not with Find Solution 1 Agent. One of the outputs of the GAIA analysis is a Preliminary Interaction Model and these interactions will be represented in that model. For that representation, we will use AUML as a notation. That model is finished during design but we are not presenting that information in this paper.

Finally from this diagram we can also see the access to the environment that each agent will do. The environment is represented by the dark large ellipsis and each resource by a small and darker ellipsis inside the environment. We did not represent all the resources involved, so that the diagram did not become too cluttered. The arrows between the agents and the resources represent the permissions that the agents/roles have. These permissions are taken from the Preliminary Role Model that is also one of the outputs of the analysis. The Preliminary Role Model is finished during design, but, from this diagram it is possible to see that some agents/roles have Read permission, that is, are only authorized to read or sense from the environment. Others have Change permission. In this latter case, they are authorized to change the environment.

We would like to point out that in the final design of our MAS architecture we will have to take into account issues like, computational time needed by each agent to perform it's tasks, deployment in a distributed environment and communication inside the MAS, security, privacy, etc. These issues might lead to several changes in this architecture.

We know that the information presented in this paper is a simplified and yet incomplete version of the system we are developing. However we believe that it has the necessary information to motivate us in applying a multi-agent system, with all the "intelligent" advantages that come from that kind of systems, to such a demanding domain of business. Usually these kinds of problems are approached with methods from Operations Research. From our experience and even though there are some commercial applications to deal with this domain, the fact is that they all use the same type of algorithm or method to solve all kind of problems raised during airline operations. That characteristics leads to the fact that some of the solutions for the problems still have to be found by the user, using his/her experience of the domain. We believe that a multi-agent system, with the "intelligent" features we are incorporating in our system, will have all the conditions to be the tool for the solution of all type of problems that might appear.

6 Future Work

Besides continuing to work in our MAS, we would also like to address the topic of cooperation between airlines in recovery operation. We are working in a definition of a Electronic Market of crew members and aircrafts, that will contribute to the cooperation between airlines companies in this area. Nowadays, airline companies, some times have to rent aircrafts and crew members as a "bundle" from other companies, to be able to solve problems. What we want to do is to create a market of crew members and/or aircrafts that could be used by companies to solve their on problems. For example, if a specific airline company needs a crew member to do a specific duty, the best solution (or the only one in some cases) might be the use of a crew member from a different airline. We know that things are not so linear when regarding the use of crew members from other airlines companies. Airline internal rules, labor and country specific rules, European rules and crew qualifications, just to mention a few, are to be taken into account when approaching this subject. Others things like the communication between different information systems, security, privacy and authorization, are also problems that we will have to deal with. Finally we will also have to deal with everything that characterizes an electronic market. We know that it is a hard job but we believe that, after solving all the questions related to the regulation that will be necessary, this kind of market might be a profitable one for airline companies or to companies that might be created specialized in supplying solutions to these problems.

Acknowledges

António Castro is grateful to TAP Portugal for the support during this research and for allowing the use of real data extracted from the information system used in TAP.
Both authors are also grateful to IBM University program for allowing the use of several tools for the development of this system.

References

[1] Cynthia Barnhart, Peter Belobaba, and Amedeo R. Odoni. Applications of operations research in the air transport industry. *Transportation Science*, 37(4):368–391, 2003.

[2] B. Bauer, J. P. Muller, and J. Odell. Agent uml: A formalism for specifying multiagent software systems. *International Journal of Software Engineering and Knowledge Engineering*, 11(3), 2001.

[3] Bernhard Bauer and James Odell. Uml 2.0 and agents: How to build agent-based systems with the new uml standard. *Journal of Engineering Applications of Artificial Intelligence*, 18(2):141–157, 2005.

[4] P. Bresciani, P. Giorgini, F. Giunchiglia, J. Mylopoulos, and A. Perini. Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems*, 8(3):203–236, 2004.

[5] J. Castro, M. Kolp, and J. Mylopoulos. Towards requirements-driven information systems engineering: The tropos project. *Information Systems*, 27(6):365–389, 2002.

[6] Jens Clausen, Allan Larsen, and Jesper Larsen. Disruption management in the airline industry - concepts, models and methods. Technology Report IMM-Technical Report-2005-01, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 2005.

[7] TAOM4E Tool for Agent Oriented visual Modeling for the Eclipse Platform. http://sra.itc.it/tools/taom4e/, 11 2005.

[8] Java Agent Development Framework. http://jade.tilab.com.

[9] Michael N. Huhns. Bootstrapping cognition through language. In *Proceedings of the Third International Workshop on Agent-Oriented Methodologies, Vancouver, Canada, October 2004*, 2004.

[10] N. Kohl and S. Karisch. Airline crew rostering: Problem types, modeling and optimization. *Annals of Operations Research*, 127(0):223–257, 2004.

[11] N. Kohl, A. Larsen, J. Larsen, A. Ross, and S. Tiourline. Airline disruption management - perspectives, experiences and outlook. Technology Report CRTR-0407, Carmen Research, September 2004.

[12] Java From Sun Microsystems. http://java.com.

[13] J. Mylopoulos, L. Chung, and E. Yu. From object-oriented to goal-oriented requirements analysis. *Communications of the ACM*, 42(1):31–37, 1999.

[14] TAP Portugal. http://www.tap.pt.

[15] IBM Rational Software. http://www.ibm.com/rational.

[16] JF. Zambonelli, N. Jennings, and M. Wooldridge. Developing multi-agent systems: The gaia methodology. *ACM Transactions on Software Engineering and Methodology*, 12(3):317–370, 2003.